S.N. 09/652365

FIG. 1

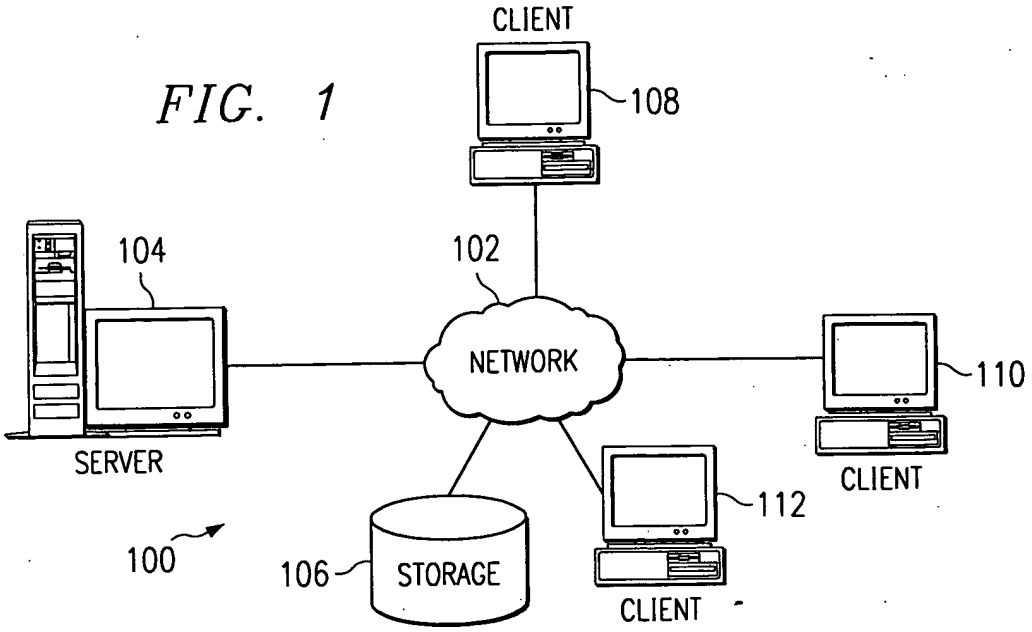
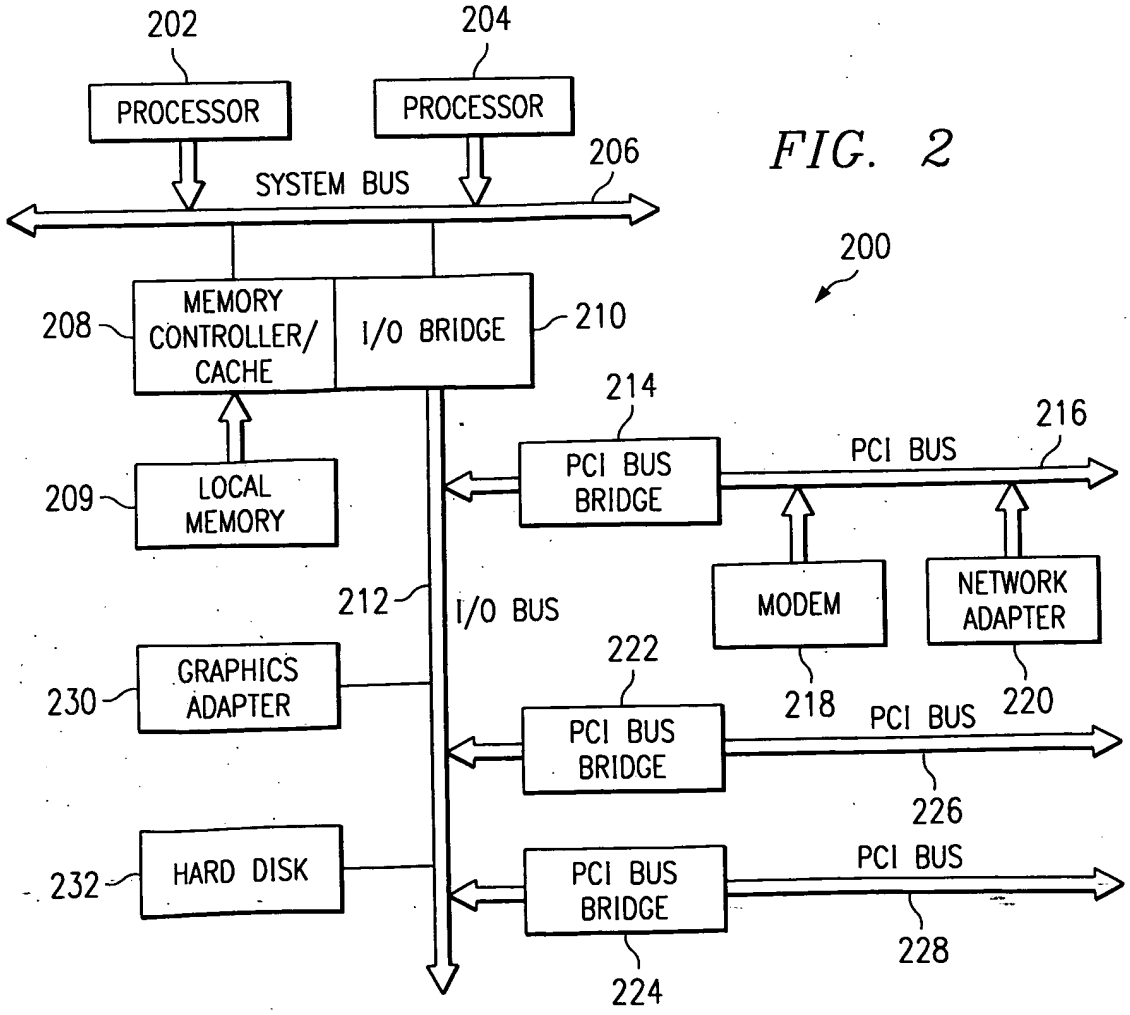


FIG. 2



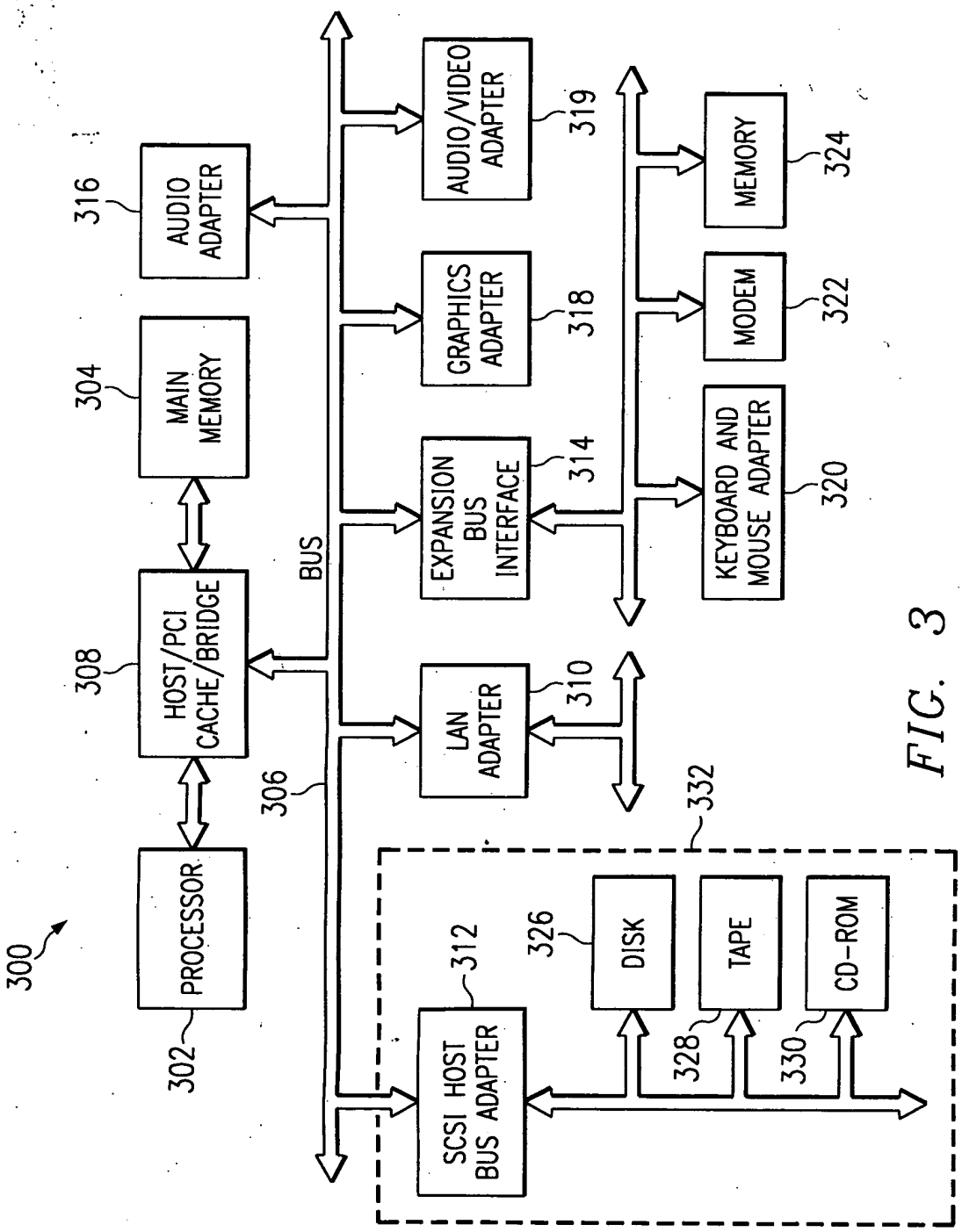
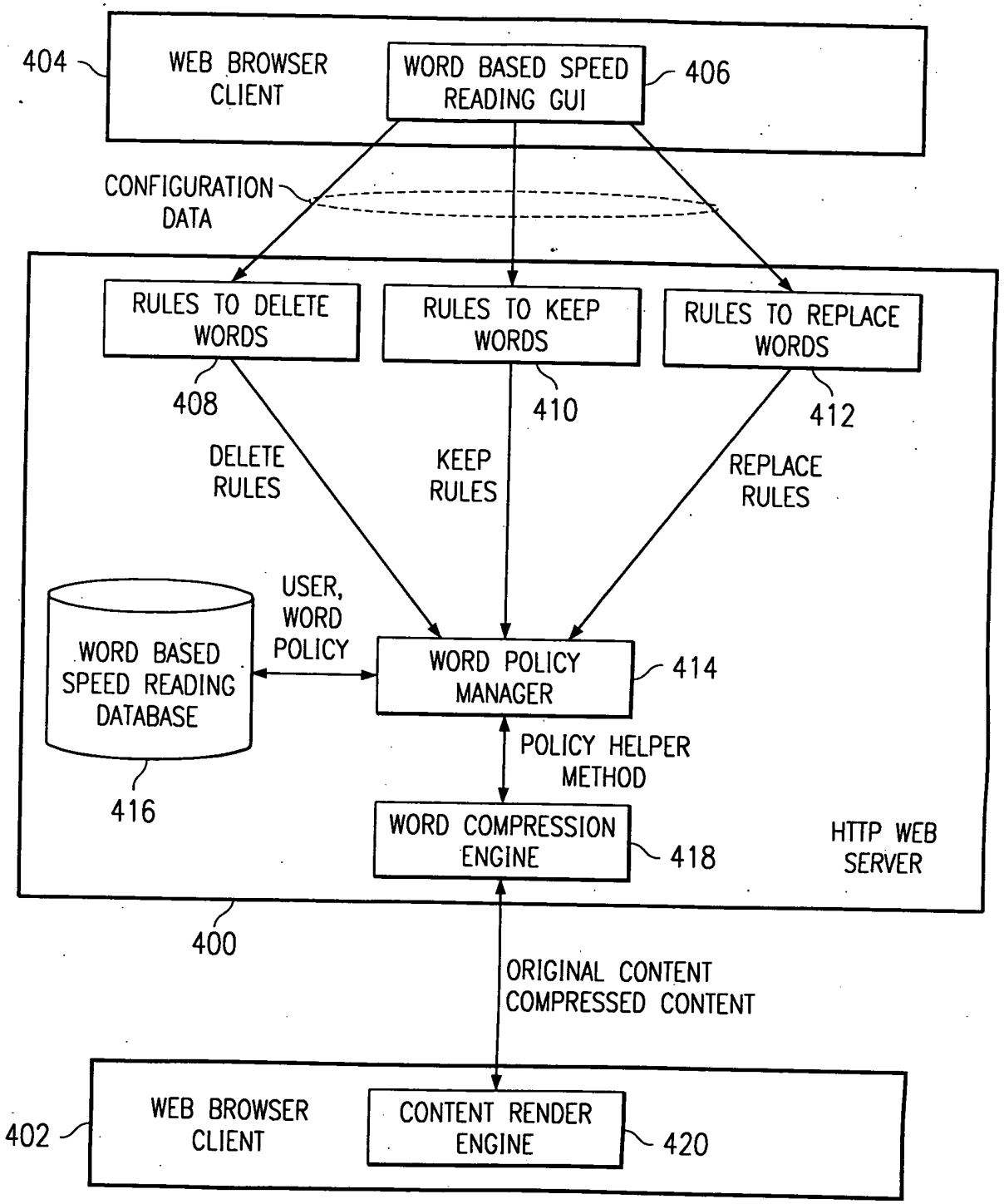

300
PROCESSOR
302
HOST/PCI CACHE/BRIDGE
308
MAIN MEMORY
304
AUDIO ADAPTER
316
BUS
306
LAN ADAPTER
310
EXPANSION BUS INTERFACE
314
GRAPHICS ADAPTER
318
AUDIO/VIDEO ADAPTER
319
KEYBOARD AND MOUSE ADAPTER
320
MODEM
322
MEMORY
324
332
SCSI HOST BUS ADAPTER
312
DISK
326
TAPE
328
CD-ROM
330

FIG. 3

FIG. 4


404
WEB BROWSER CLIENT
WORD BASED SPEED READING GUI
406
CONFIGURATION DATA
RULES TO DELETE WORDS
RULES TO KEEP WORDS
RULES TO REPLACE WORDS
408
410
412
DELETE RULES
KEEP RULES
REPLACE RULES
USER, WORD POLICY
WORD BASED SPEED READING DATABASE
WORD POLICY MANAGER
414
POLICY HELPER METHOD
416
WORD COMPRESSION ENGINE
418
HTTP WEB SERVER
400
ORIGINAL CONTENT
COMPRESSED CONTENT
402
WEB BROWSER CLIENT
CONTENT RENDER ENGINE
420

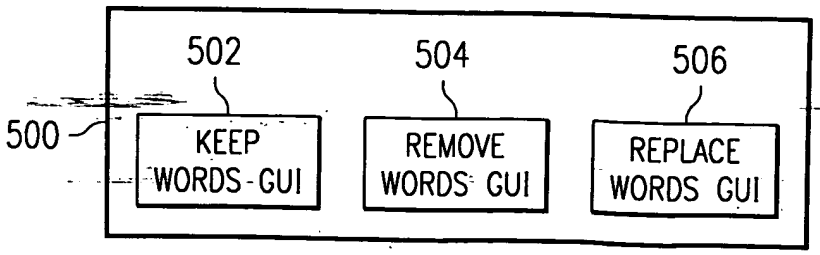
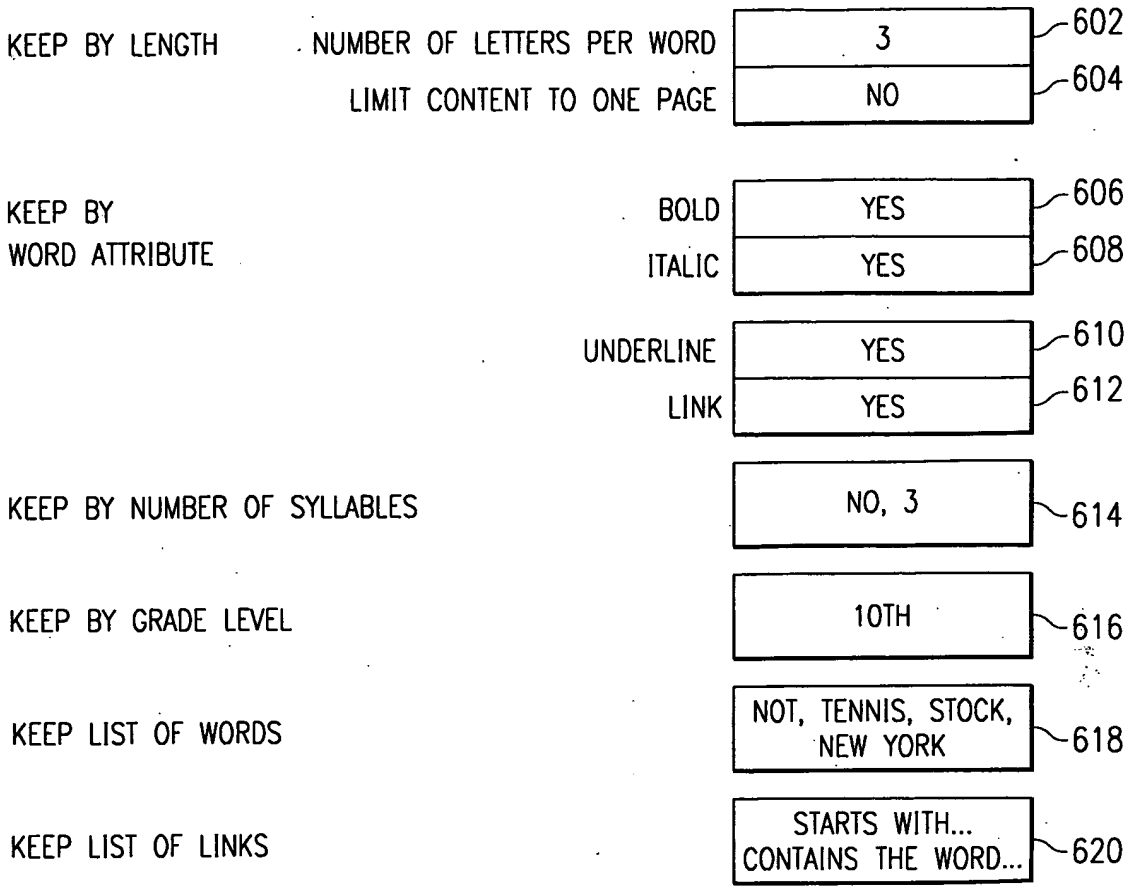

500
502
504
506
KEEP WORDS-GUI
REMOVE WORDS GUI
REPLACE WORDS GUI

FIG. 5

FIG. 6


KEEP BY LENGTH
NUMBER OF LETTERS PER WORD
3
602
LIMIT CONTENT TO ONE PAGE
NO
604
KEEP BY WORD ATTRIBUTE
BOLD
YES
606
ITALIC
YES
608
UNDERLINE
YES
610
LINK
YES
612
KEEP BY NUMBER OF SYLLABLES
NO, 3
614
KEEP BY GRADE LEVEL
10TH
616
KEEP LIST OF WORDS
NOT, TENNIS, STOCK, NEW YORK
618
KEEP LIST OF LINKS
STARTS WITH... CONTAINS THE WORD...
620

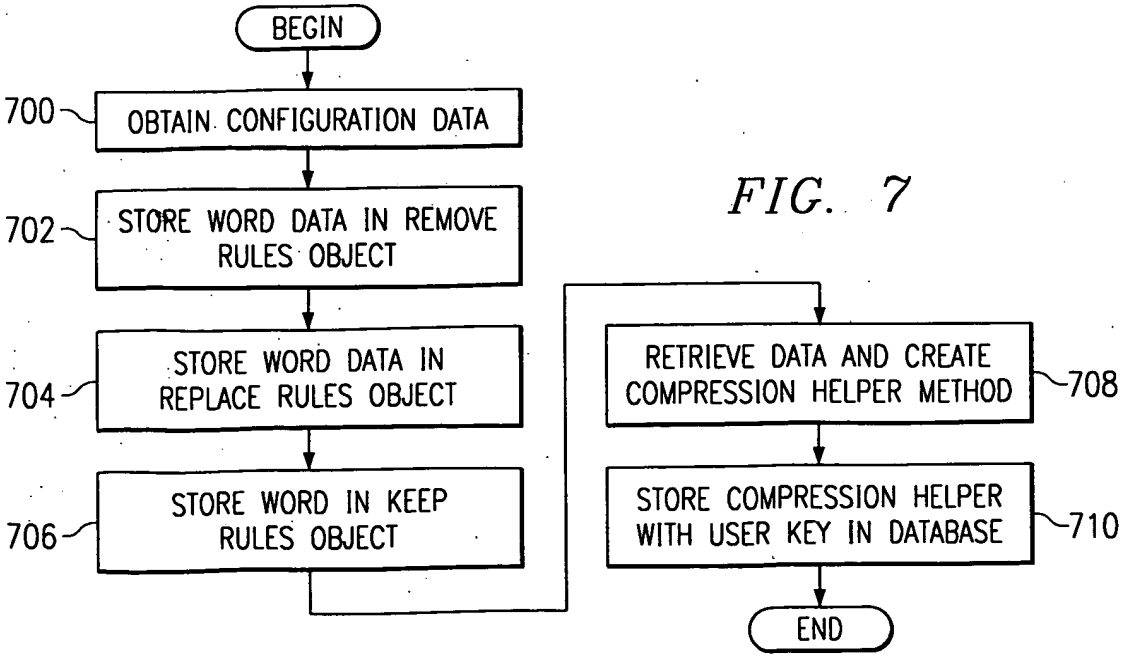

BEGIN
700
OBTAIN CONFIGURATION DATA
702
STORE WORD DATA IN REMOVE RULES OBJECT
704
STORE WORD DATA IN REPLACE RULES OBJECT
706
STORE WORD IN KEEP RULES OBJECT
RETRIEVE DATA AND CREATE COMPRESSION HELPER METHOD
708
STORE COMPRESSION HELPER WITH USER KEY IN DATABASE
710
END

FIG. 7

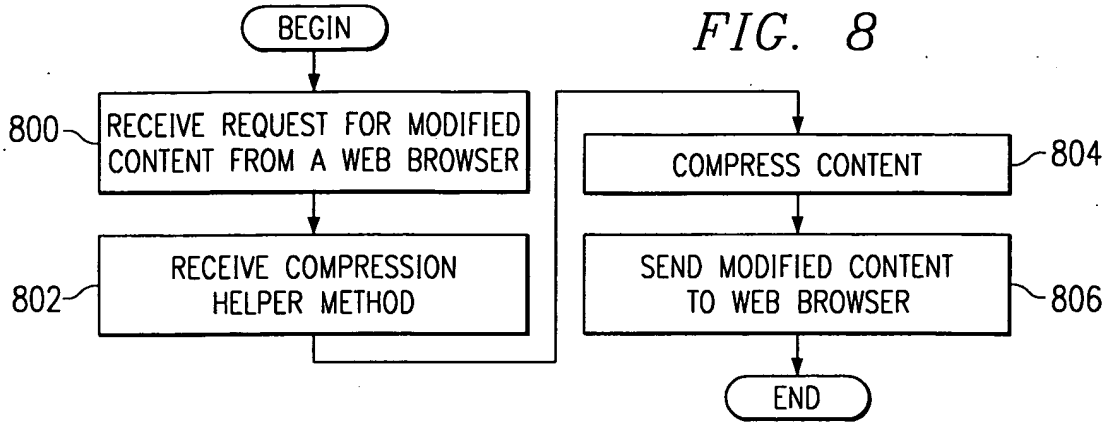


FIG. 8

FIG. 9A

900

```
DeleteWord { // used if delete words defined in GUI

//init constructor
  DeleteWord (int length, int syllable, int Difficulty, int attribute)

  //data
  int byLength;
  int bySyllable;
  int byDifficulty;
  int byAttribute;
  Vector removeFullWords;
  Vector removeStartsWithWords;

  //methods
  void setFullWord (String word) { }
  void setStartsWith (String word) { }
  boolean delete Word (String word) {
    //compare length, difficulty, attrributes
    //compare with removeFullWords list
    //compare with removeStartsWithWords list
    //return true or false

void setWord (String word) {
//used by the GUI to add words to delete (or extended by Keep Class below)
//add to Vector
}
void setStartsWithWord (String word) {
//used by-the GUI to add words
//add to Vector
}
```

FIG. 9B

```
ReplaceWords { // used if replace words defined in GUI
  //data
  Hashtable wordsToBeReplaced
       //key=word to be replaced
       //value=replacement word
  //methods
  boolean replace (String word) {
    //check hash to decide return true
}
  void setReplacement (String wordToBeReplace, replacement) {
//used by the GUI to add words
}
  String getReplacement (String word) { }
}
KeepWords { // used if keep words defined in GUI
     extends DeleteWords

     //init constructor
     KeepWords (int length, int syllable, int Difficulty, int attribute)

//methods

boolean keepWord (String word) {
       //compare length, difficulty, attributes
       //compare with removeFullWords list
       //compare with removeStartsWithWords list
       //return true or false.

       }

}
```

FIG. 10

1000

```
CompressionHelper  {

//methods
String getReplacement (word){ };

   int getNumSyllables (String word) { return numberOfSyllables }
   int getWordLength (String word) { return wordLength )
   int getDifficulty (String word) { return GradeLevelDifficulty }
   int getAttributes (String word) //bold=1, underline=2, italic=3, etc.
     { return VectorOfAttributes

//data
boolean isPartOfWordRemoveList = DeleteWords.deleteWord (word);
boolean isPartOfWordKeepList = KeepWords.keepWord (word);
boolean isPartOfWordReplaceList = ReplaceWords.deleteWord (word);

//init Constructors
CompressionHelper(String word) { };
CompressionHelper(String [ ] words) { };

}
```

FIG. 11

1100

```
//String getModifiedContent (String OriginalContentInFileFormat) {
    //create InputStream from OrginalContentIFile
    //create InputStream for ModifiedContentOutputFile
    //loop through all words
        ifWordOnDeleteList //create CompressionHelper classes with each word to be
analyzed in parallel with reading the unmodified file content. After caching the
compression helpers away, the boolean flags can be used to determine how the
modified content is rendered (word removed, word replaced, word remains intact).
      and NOT on isWordOnKeepList OR isWordOnReplaceList
        //delete word
        //break next word
      else IfOnKeeplist AND NOT on WordReplaceList
        //break next word
      else //OnWordReplacelist
        //replace word

    Write result to ModifiedContentOutoutFile

}
```